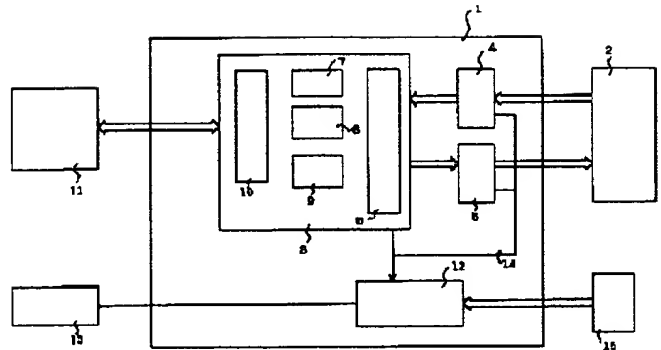


# Patent Abstracts of Japan

TITLE : ENGINE CONTROLLER





(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開平7-287603

(43) 公開日 平成7年(1995)10月31日

(51) Int.Cl.<sup>6</sup> 識別記号 庁内整理番号 F I 技術表示箇所  
G 0 5 B 15/02  
F 0 2 D 45/00 3 7 6 B  
G 0 5 B 9/02 J  
23/02 F 7531-3H  
7531-3H G 0 5 B 15/ 02 H  
審査請求 未請求 請求項の数11 O L (全 12 頁) 最終頁に続く

(21) 出願番号 特願平6-80133

(22) 出願日 平成6年(1994)4月19日

(71) 出願人 000005108

株式会社日立製作所

東京都千代田区神田駿河台四丁目6番地

(71) 出願人 000232988

日立オートモティブエンジニアリング株式  
会社

312 茨城県ひたちなか市大字高場宇鹿島  
谷津2477番地3

(72) 発明者 佐々木 正浩

茨城県勝田市大字高場宇鹿島谷津2477番地  
3 日立オートモティブエンジニアリング  
株式会社内

(74) 代理人 弁理士 小川 勝男

最終頁に続く

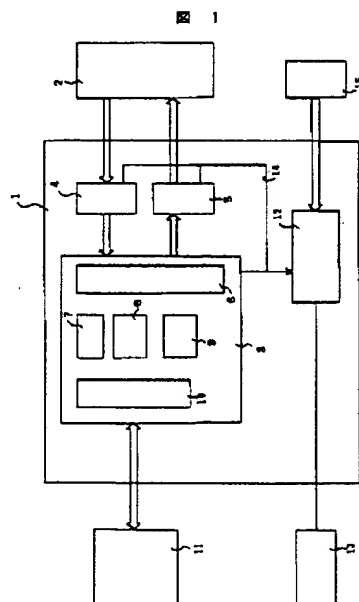
(54) 【発明の名称】 エンジン制御装置

(57) 【要約】

【目的】 本発明は、エンジン制御装置に関し、その目的は、オンボードでの不揮発性メモリ内容変更後の、初期不良を未然に摘出する装置を提供することにある。

【構成】 エンジン制御装置1の安定化電源電圧を、通常電圧よりも高くすることで達成される。

【効果】 上記構成により、オンボードでの不揮発性メモリ内容変更後のスクリーニングが可能となり、初期不良を未然に摘出できるため、市場不良を未然に防止する効果がある。



## 【特許請求の範囲】

【請求項1】データの消去及び書き込みが可能であり、このデータの消去は予め定められたブロック毎に行える読み出し専用の不揮発性メモリと、データの書き込み及び読みだし可能な揮発性メモリとを内蔵し、この不揮発性メモリに書き込まれたプログラムにしたがって動作するマイクロコンピュータを実装したエンジン制御装置において、このエンジン制御装置に外部装置と電気的に接続でき、上記外部装置からの情報に従い上記マイクロコンピュータに内蔵された不揮発性メモリの内容を消去、及びデータの書き込みができる事を特徴としたエンジン制御装置。

【請求項2】請求項1記載のエンジン制御装置において、消去及び書き込みが可能な不揮発性メモリの内容は、上記マイクロコンピュータを動作させるプログラムであることを特徴とするエンジン制御装置。

【請求項3】請求項2記載のエンジン制御装置において、上記プログラムの消去及び書き込みの実行が上記マイクロコンピュータに内蔵された揮発性メモリ上に格納されたプログラムによって実施されることを特徴とするエンジン制御装置。

【請求項4】請求項3記載のエンジン制御装置において、上記揮発性メモリに格納される消去及び書き込みのプログラムは、上記不揮発性メモリの消去及び書き込みの動作を実行する前に該不揮発性メモリに予め格納されており、前記外部装置からの情報が入力されてから上記揮発性メモリに転送されることを特徴とするエンジン制御装置。

【請求項5】請求項1記載のエンジン制御装置において、不揮発性メモリの内容の消去及び書き込みは、エンジンを制御するためのデータのみ可能とし、マイクロコンピュータを動作させるプログラムは消去及び書き込みを禁止することを特徴としたエンジン制御装置。

【請求項6】請求項5記載のエンジン制御装置において、上記不揮発性メモリの消去及び書き込みの回数を記憶する手段を有し、上記消去及び書き込みの回数が所定回数に達した場合は消去及び書き込みを禁止することを特徴とするエンジン制御装置。

【請求項7】請求項6記載のエンジン制御装置において、上記不揮発性メモリの消去及び書き込みの回数を記憶する手段は、上記消去可能なブロック毎に有していることを特徴とするエンジン制御装置。

【請求項8】請求項1記載のエンジン制御装置において、不揮発性メモリの電源電圧を可変出来ることを特徴とするエンジン制御装置。

【請求項9】請求項1記載のエンジン制御装置において、外部装置からの情報に従い、エンジン制御装置内の安定化電源の電圧値を可変出来ることを特徴とするエンジン制御装置。

【請求項10】請求項1記載のエンジン制御装置におい

て、外部装置から不揮発性メモリの電源電圧を供給出来ることを特徴とするエンジン制御装置。

【請求項11】請求項1記載のエンジン制御装置において、通常の不揮発性メモリの電源電圧よりも高い電圧を印加し、周囲温度を高温にして動作させることにより、オンボード上で不揮発性メモリのスクリーニングが出来、不揮発性メモリの初期不良を摘出出来ることを特徴とするエンジン制御装置。

## 【発明の詳細な説明】

【0001】

【産業上の利用分野】本発明は、エンジン制御装置に係り、オンボード上で不揮発性メモリを内蔵したCPUを使用したエンジン制御装置の初期不良を未然に摘出するのに好適なエンジン制御装置に関する。

【0002】

【従来の技術】従来、エンジンを制御するために、エンジン制御装置が用いられているが、上記エンジン制御装置には定められたプログラムに従い動作するCPUと上記プログラムを格納する不揮発性メモリであるROMが備えられている。このROMにはプログラムと、エンジンに供給する燃料噴射量や点火時期などを決めるデータが格納されている。近年、上記ROMを内蔵したシングルチップモード可能なCPUが発売されているが、このCPU内ROMには1回のみプログラムやデータを書き込むことが可能であり、エンジン制御装置に実装後、上記内蔵ROMに書き込む方法等も発明されている。

【0003】上記のように、基板に実装後CPUに内蔵したROMに書き込みを行う技術として特願平2-20416号公報がある。

【0004】

【発明が解決しようとする課題】上記従来技術ではCPUを基板に実装後内蔵ROMに書き込みは可能であるが、初期不良を未然に摘出することを考慮していない。即ち、エンジン制御装置を市場に供給後、不具合が発生するという問題があった。

【0005】本発明の目的は、上記した従来技術の欠点を除いたエンジン制御装置を供給することにある。

【0006】

【課題を解決するための手段】上記目的は、オンボード上に不揮発性メモリの電源電圧を、通常よりも高い電圧にして印加し、周囲温度を高温にして動作させ、スクリーニングを実施することにより達成される。

【0007】

【作用】上記構成により、オンボード上に不揮発性メモリのスクリーニングが可能となり、初期不良を未然に摘出できるという効果がある。

【0008】

【実施例】以下、本発明の一実施例を図面を用いて説明する。

【0009】図1は本発明の構成を示すブロック図であ

る。1はエンジン制御装置。2は上記エンジン制御装置1によって制御されるエンジン。上記エンジン2には図示していないが運転状態を表す様々なセンサが取り付けられており、上記センサからの信号がエンジン制御装置1に入力される。エンジン制御装置1は上記センサからの信号を波形処理する入力処理回路4と、このセンサ信号を入力しエンジン2の最適運転状態を演算するCPU3と、上記演算された結果を制御信号として受け、図示していないがエンジン2に取り付けられている燃料噴射装置や点火装置などのアクチュエータを駆動する出力回路5から構成されている。上記CPU3には制御用プログラムに従い動作するMPU7と、このMPU7を動作させるプログラムを格納するROM8と、MPU7の演算結果を格納するRAM9と、前記入力処理回路4からの信号を受け、また、出力回路5に制御信号を出力するI/O6と、外部機器とのデータ通信用の通信回路10が内蔵されている。上記エンジン制御装置1には外部機器であるメモリ書換機11と電気的に接続可能であり、CPU3内の通信回路10に接続される。安定化電源回路12では、バッテリー13を電源として、常に最適な一定の電圧を電源ライン14により各部に供給している。また、この電圧は、外部の安定化電源電圧切り換え信号発生器15からの信号により可変出来る回路構成になっている。ここで、上記ROM8は特別な条件下で内容の消去及び書き込みができるメモリである。

【0010】図2は上記CPU3のメモリマップを示す。図2(a)はCPU3全体のメモリマップであり、3aはROM8の領域、3bは外部メモリ空間、3cはRAM9の領域、3dはMPU7の内部レジスタ領域である。(b)は上記ROM8内のブロック構成を表す。ROM8内はいくつかのブロックに分かれており、図2(b)の場合はA~Pの16ブロックに分割されており、ROM8内の内容は各ブロック毎に消去可能な構成である。エンジン制御装置1で使用しているROM8内の構成は大別すると制御プログラムが格納されているプログラム領域と制御データが格納されている領域及び消去不可能領域とに分かれている。本実施例では、ブロックA~Kがプログラム領域、ブロックL~Pまでがデータ領域である。図3は上記ROM8内のデータを消去するための専用レジスタの構成を示す。消去レジスタは上記内部レジスタ領域3dに配置され、2バイトで構成されている。消去レジスタ1及び2の各ビットはROM8内の各ブロックA~Pに対応しており、このビットを例えば1にするとそれに対応したブロックの内容が消去される。従って、例えば、データ領域であるブロックMのデータのみを書き換えたい場合は、まず、消去レジスタ2のビット4を1にしてブロックMのデータを消去し、その後ブロックMにデータを書き込む操作を行えば良い。

【0011】図4は外部機器であるメモリ書換機11との通信でCPU8内のメモリ内容を書き換えていくとき

のCPU8の内部状態を示す状態遷移図である。メモリ書換機11との通信を開始する前はS1でメモリ書換機11からの通信待ち状態にある。ここで、メモリ書換機11から「INIT」と言うコマンドが入力された場合、〈INIT〉と言うコマンドをエンジン制御装置1からメモリ書換機11に送信してやり、「INIT」コマンドを受信したことを知らせると同時に、S2の設定コマンド待ち状態にはいる。S1で「INIT」以外のコマンドが入力された場合は〈エラー〉コマンドを送信し、元の状態S1に戻る。以下、各状態においても同様に、その状態に応じたコマンドが入力された場合はそれに対するコマンドを送信し、次の状態に移移する。それ以外の場合は〈エラー〉コマンドを送信し、初期状態であるS1に戻る構成である。初期状態S1から「INIT」コマンド入力後、S2の設定コマンド待ち状態に移移し、次のコマンドを待つ。S2でプログラム変更モードにはいるための「PROG」コマンドを受信した場合はS3に、データ変更モードにはいるための「DATA」コマンドを受信した場合はS7に、それ以外はS1にそれぞれ移移する。S3に移移した場合はプログラム変更モードに入り、S7に移移した場合はデータ変更モードに入る。S3で「GO」コマンドを受信した場合はS4に入り、変更のデータが入力された場合S5に移り、データ変更の処理を行う。この処理が終了したら次のデータが入力されるのを待つため、〈データ〉コマンドを送信後、再びS4に戻る。所定個数のデータが入力されたら、S5から終了コマンド待ち状態のS6に移る。S6では「EXIT」コマンドが受信された場合、〈EXIT〉コマンドを送信し、S1に戻ってプログラム変更モードを終了する。一方、S2で「DATA」コマンドを受信した場合はデータ変更モードに入り、S7で変更する上記ROM8内のブロックの指定である「ADDR」データを受信し、S8に入る。S8で処理の開始コマンドである「GO」コマンドを受けた後、S9とS10でデータ変更の処理を行う。S9では変更すべきデータが受信されたらS10に移り、変更の処理を実行する。処理終了後、〈データ〉を送信し、S9の状態に再び戻り、次のデータを待つ。所定個数のデータが入力されたらS6の状態に移り、「EXIT」コマンド待ち状態にはいる。以下、プログラム変更モードと同様に「EXIT」コマンドが入力されたら、〈EXIT〉コマンドを送信し、S1の状態に戻ってデータ変更モードを終了する。

【0012】図5は図4の通信を行ったときのエンジン制御装置1内のプログラムの動作を示すフローチャートである。本実施例ではメモリ書換機11との通信はシリアル通信を用いている。図5はメモリ書換機11からデータが送信されたとき発生するSCI割り込みの処理を示している。SCI割り込みが発生したら、まずS20でINIT flag, S21でPROG flag, S22でD

ATA flagが1かどうかを確認する。S20でINIT flagが0の場合、現在図4のS1である通信待ち状態にあることを示しており、S28で今回入力されたデータが「INIT」コマンドであるかどうかを確認する。

「INIT」コマンドでない場合はS25に進み、〈エラー〉コマンドを出力し、INIT、PROG、DATAの各flagをそれぞれ0にして通信の初期状態に戻る処理をしてSCI割り込み処理を終了する。S28で「INIT」コマンドであることを確認したら、S29で〈INIT〉コマンドをメモリ書換機11に出力し、且つINIT flagを1にして処理を終了する。S20でINIT flagが1であった場合、図4のS2以降の状態にあることを示しているため、以下でどこ状態であるかを確認する。S21ではPROG flagでプログラム変更モードかを確認し、flagが1のときはS31に進み、プログラムの変更処理を行う。そうでないときは、S22でDATA flagでデータ変更モードかを確認する。データ変更モードである場合はS30に進み、データ変更処理を行う。S30でflagが0の場合は、図4のS2の状態にあるため、今回入力されたデータが「PROG」または「DATA」コマンドであるかをS23、S24で確認する。S23で「PROG」コマンドが入力されたと判断されたら、S27に進み、〈PROG〉コマンドを出力及びPROG flagを1にしてプログラム変更モードに入る。一方、S24で「DATA」コマンドが入力されたと判断されたらS26に進み、〈DATA〉コマンドの出力及びDATA flagを1にしてデータ変更モードに入る。S23、S24でどちらのコマンドも確認されなかった場合はデータ通信エラーと判断してS25に進み、〈エラー〉コマンドの出力及びINIT、PROG、DATAの各flagを0にして初期状態である図4のS1の状態に戻る。

【0013】図6は図5の処理でプログラム変更モードに入ったときのS31の処理を説明するフローチャートである。CPU3は通常内蔵したROM8に格納されたプログラムで動作するため、そのプログラムの書き換えを行う際、ROM8の内容を消去するとCPU3が動作しなくなってしまう。それを防ぐため、プログラムの変更を行うときはプログラム書換のプログラムをROM8上で動作させるのではなく、RAM9上で行わせる必要がある。図6の処理はそのプログラム書換のプログラムをRAM9上で行わせる様にした処理を示す。尚、プログラム書換のプログラムは元々ROM8上にあり、それを実行する際は一旦RAM9に転送後行うようにしている。図6のS40で「GO」コマンドかどうかを確認し、もし、「GO」コマンドでなければ、送信されたコマンドが異常であるため、S43に進み〈エラー〉コマンド出力後、INIT、PROGの各flagを0にして初期状態に戻る。「GO」コマンドであればS41へ進み、プログラム書換のプログラムをROM8からRAM

9へ転送する処理を行う。転送処理が終了後、S42でRAM9に転送したプログラムへジャンプし、RAM9上のプログラムを実行する。

【0014】図7は上記処理でRAM9上に転送されたプログラム書換のプログラムの処理を示すフローチャートである。図6のS42でRAM9にジャンプした後S50に進み、ROM8のプログラム領域であるブロックA〜Kの内容を消去する。ここで、メモリ書換機11からシリアル通信でデータを送信してきたとき、SCI割り込みがCPU3に発生するが、そのときの処理プログラムにジャンプするためにベクタアドレスが必要となる。ベクタアドレスにはその割り込み処理のプログラムの先頭アドレスが入っている。本実施例の場合、上記ベクタアドレスはROM8のブロックA上に配置されており、上記プログラム書換のプログラムを実行するときは、そのベクタアドレスをRAM9にジャンプするように設定する必要がある。また、プログラムの書換が終了した後、SCI割り込みが発生したときはROM8上のプログラムを実行させる必要があるため、再度上記ベクタアドレスを書き換えなければならない。S51ではRAM9上でプログラムを実行させるため、ROM8上のブロックAにあるベクタアドレスに上記RAM9上のプログラムの先頭アドレスを書き込む処理を行う。この処理を終了することによりプログラム書き込み可能となる。以下、S52で〈GO〉コマンドを出力し、GO flagを1にしてプログラム書換中であることを記憶する。S53では、プログラム書換の際、順次送信されてくるデータを書き込むアドレスを表すMADRをイニシャライズして、一旦処理を終了して図4のS4のデータ待ち状態になる。この状態でデータが送信されて来たとき再びSCI割り込みが発生し、図7のS54に進む。S54ではプログラム書き込み中かどうかをGO flagで確認し、書き込み中と判断された場合はS55に進み、現在書き込もうとしているアドレスがブロックAかどうか確認する。ブロックAと判断された場合は、入力されたデータを一旦RAM9上のMPRGと言う領域に格納する。MPRGはブロックAと同じだけのメモリサイズを有し、送信されてきたデータをアドレス順に順次格納していく。これは、ブロックAには上記のように、SCIのベクタアドレスがあるため、直接ROM8上に書き込みを行うとこのベクタアドレスの内容を破壊してしまうため、それを防止するための処置である。一方、S55でブロックAに相当していないときはS56でMADRにあるROM8上のアドレスに転送されたデータを書き込む。次にS58で書き込みアドレスを示すMADRの内容をインクリメントする。これで、送信されたデータは所定のアドレスにき込まれたことになり、次のデータが入力されるまで一旦処理を終了するため、S59で〈データ〉コマンドを出力する。S60ではMADRの値が所定値以上、即ち、プログラム領域に全て書き込ん

だかどうかを確認する。MADRが所定値以下であれば、まだプログラム変更中ということで次のデータが入力されるまで割り込み処理を終了する。S60で所定値以下と確認された場合はプログラム領域の変更が全て終了したと判断し、GO flagを0にしてプログラム変更処理が終了したこと記憶する。この後、SCI割り込みが発生した場合、S54でGO flagが0であるため、送信されてきたデータが「EXIT」コマンドであるかどうかをS62で確認する。「EXIT」コマンドでなければ、そのコマンドが入力されるまでまつため、一旦処理を終了する。S62で「EXIT」コマンドを確認したらS63以降に進み、上記ブロックAの分をRAM9のMPRGから転送する処理を行う。まず、S63でブロックAの内容を消去し、S64で上記ブロックAにMPRGの内容を書き込む。この書き込み処理が終了したらS65で「EXIT」コマンドを出力し、INT、PROGの各flagを0にして処理を終了する。以上の動作でROM8内のプログラムが変更できる。

【0015】図8はエンジン2の制御データを変更するときの処理を示すフローチャートである。データ変更処理は、上記プログラム変更処理とは異なり、変更プログラムをRAM9上で動作させる必要はない。図5の処理でデータ変更モードと判断され、S30に進んだら図8のS70の処理を実行する。S70では変更するROM8上のブロックを示すデータが入力済みであることを表すADDR flagが1かどうかを確認する。ADDR flagが0の場合、今回入力されたデータが上記変更対象のブロックを表すデータであるため、S71でRAM9上のM1にそのデータを格納する。次にS72でそのブロックが書換可能なブロック、即ち、指定されたブロックがデータ領域のブロックであり、プログラム領域のブロックではないことを確認する。もし、データ領域ではないとき、S74に進み、〈エラー〉コマンドの出力とINIT、DATAの各flagを0にして、初期状態に戻る。S72でデータ領域であることを確認後、S73でそのブロックが書き換えられた回数を記憶しているDCCNT1の値が所定値以上であるかどうかを確認する。これは、書換可能な不揮発性メモリの場合、その構造上書き込み回数に制限がある。この制限を超えて書き込みを行った場合、素子の破壊につながる可能性がある。本発明ではこれを防止するため、書き込み回数を各ブロック毎に記憶しておき、制限回数を超えている場合は書き込みを禁止している。S73で所定書き込み回数を超えていると判断された場合はエラーとして、S74に進む。S73で所定書き込み回数に達していない場合はS75で書き込み回数のカウント値をインクリメントする。S76で〈ADDR〉コマンドを出力し、ADDR flagを1にして次のコマンドを待つ。再びSCI割り込みが発生したとき、S70でADDR flag=1となるため、S77に進む。ここではデータ変更中であることを表すGO flag

の状態を確認し、データ変更中でなければS84で今回送信されてきたデータが「GO」コマンドであることを確認する。「GO」コマンドであればS85で上記M1に示されたブロックの内容を消去し、S86で書き込みの際のアドレスを示すMADRをイニシャライズする。更に、S87で〈GO〉コマンドを出力し、GO flagを1にする。S77でGO flagが1即ちデータ変更中と判定されたら、以降送信されてくるデータは変更用のデータなので順次指定されたアドレスに書き込んでいく。S78ではMADRに示されたアドレスに送信されてきたデータを書き込み、S79ではそのアドレス(MADR)をインクリメントする。S80ではデータの書き込みが完了したことをメモリ書き込み機11に知らせるため、〈データ〉コマンドを出力する。S82ではMADRの値が所定値に達したか、即ち、指定されたブロックのデータが全て書き換えられたかを判断し、全て書き換えられたときはS83でGO flagを0にしてデータ変更処理終了を示して割り込み処理を終了する。次にデータが送信された場合は、S84でGO flagが0であるため、S84に進む。ここで、「GO」コマンドは既に入力されているため、S88に進む。ここでは送信されてきたデータが「EXIT」コマンドかどうかを確認し、「EXIT」コマンドであれば〈EXIT〉コマンドを出力(S89)し、そうでない場合は〈エラー〉コマンドを出力(S90)してそれぞれS91に進む。S91では一連の処理を終了したため、初期状態に戻るのにINIT、DATA、ADDRの各flagをクリアしてデータ変更の処理を終了する。

【0016】図9はROM8を内蔵したCPU3のスクリーニングをオンボード上(エンジン制御装置1に実装した状態)で実施する手順を示すフローチャートである。上記方法によりROM8にプログラムまたはデータを書き込んだエンジン制御装置をS93により通常の電源電圧とし、S94で動作確認を行う。S95で正常に動作しているか否かを判定し、誤動作した場合はS104で不合格品となる。正常の場合はS96で安定化電源電圧を通常よりも高い電圧に切り換える。S97で電源電圧が切り変わったか否かを判定し、電圧が切り換わっていない場合は、S96に戻る。切り換わった場合はS98において、周辺温度を高温にした状態でCPU3にあらかじめ定めた所定時間通電し、スクリーニングを行う。スクリーニング終了後、S99において安定化電源電圧を通常電圧に切り換える。

【0017】S100で電源電圧が切り変わったか否かを判定し、電圧が切り換わっていない場合はS99に戻る。切り換わった場合はS101で動作確認を行う。S102で正常に動作しているか否かを判定し、誤動作した場合はS104で不合格品となり、正常の場合は合格となる。

【0018】以上の方法で、ROM8上のプログラムま

たはデータを外部機器からの通信で変更後、オンボード上でCPU3のスクリーニングが可能となり、初期不良の摘出が未然に行える様になる。

【0019】

【発明の効果】本発明によれば、エンジン制御内のメモリ内容変更後、オンボード状態でCPUのスクリーニングが可能となり、初期不良の摘出が未然に行える様になる。

【図面の簡単な説明】

【図1】本発明の構成を示すブロック図である。

【図2】CPU内蔵のROMの構成を示すメモリマップである。

【図3】ROM内のブロック毎に消去するレジスタの構成を示す構成図である。

【図4】メモリ交換機との通信を行ったときのCPU内の状態を示す状態遷移図である。

【図5】プログラム変更モードまたはデータ変更モード

に移るときの動作を示すフローチャートである。

【図6】プログラム変更モード時、プログラムをRAM上に転送する動作を示すフローチャートである。

【図7】RAM上で動作するプログラム変更のプログラムの内容を示すフローチャートである。

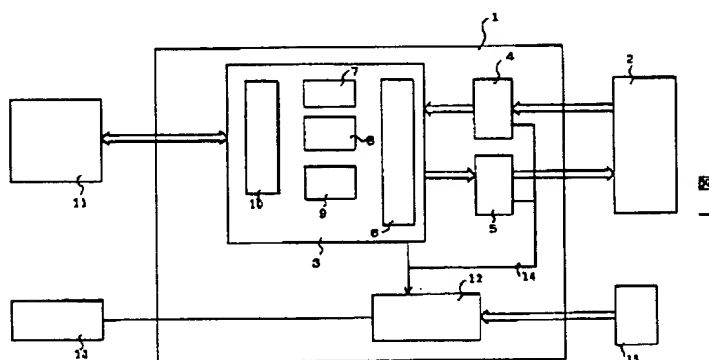
【図8】データ変更時のプログラムの内容を示すフローチャートである。

【図9】CPUのスクリーニングを行う手順のフローチャートである。

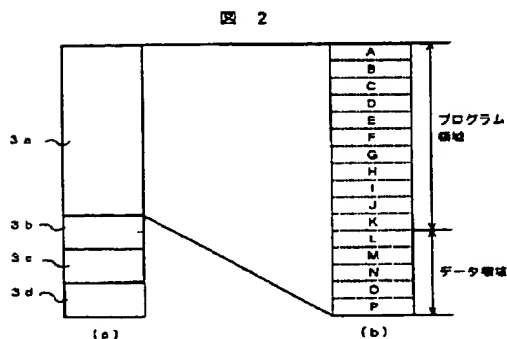
10 【符号の説明】

1…エンジン制御装置、2…エンジン、3…CPU、3a…ROM領域、3b…外部メモリ空間、3c…RAM領域、3d…内部レジスタ領域、4…入力処理回路、5…出力回路、6…I/O、7…MPU、8…ROM、9…RAM、10…通信回路、11…メモリ交換機、12…安定化電源回路、13…バッテリー、14…電源ライン、15…安定化電源電圧切り換え信号発生器。

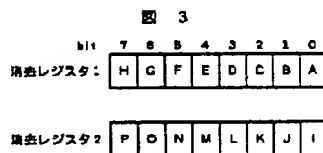
【図1】



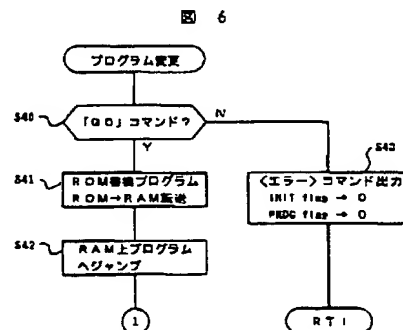
【図2】



【図3】



【図6】





(7)

特開平7-287603

【図4】

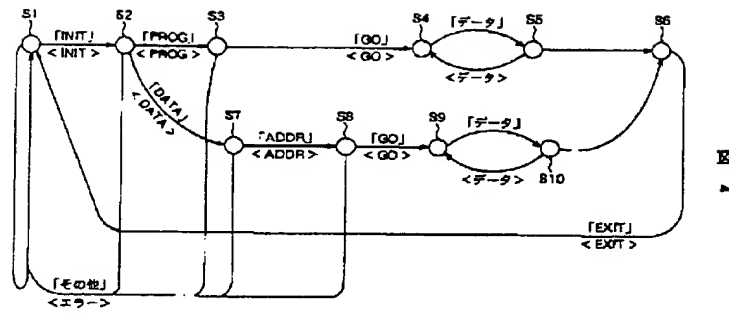
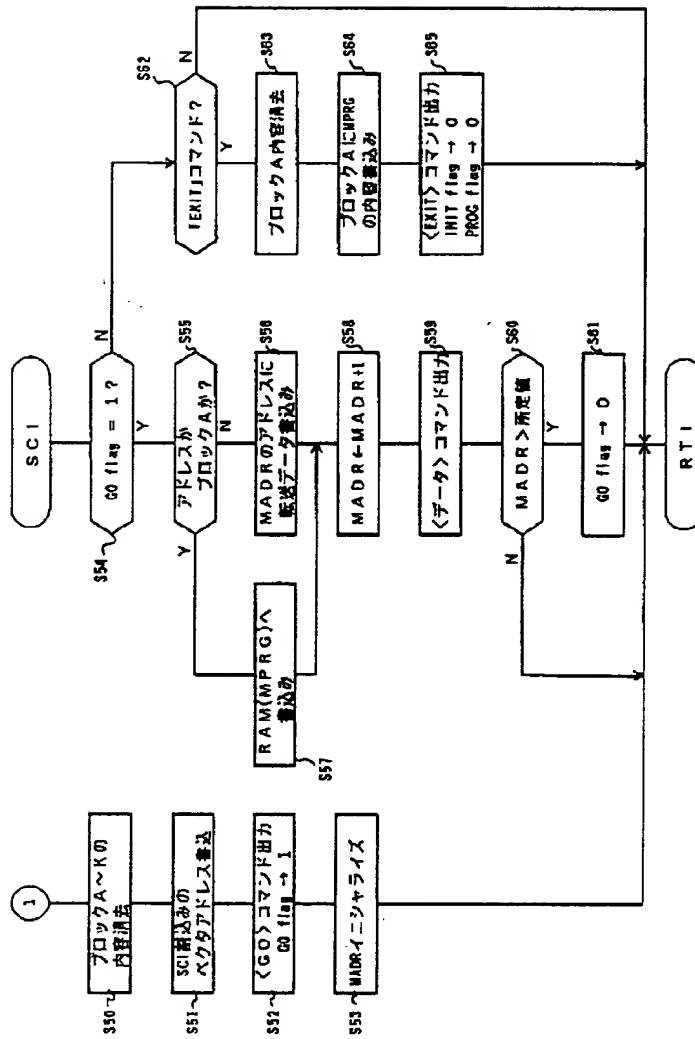


图 5

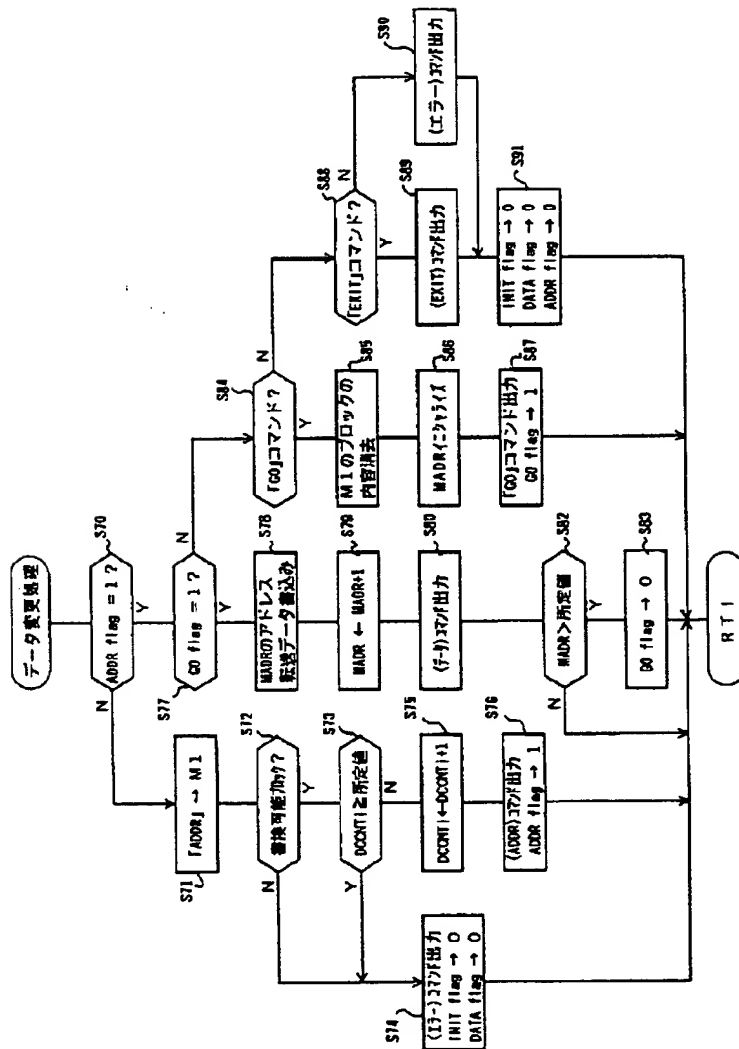


【図7】

図 7

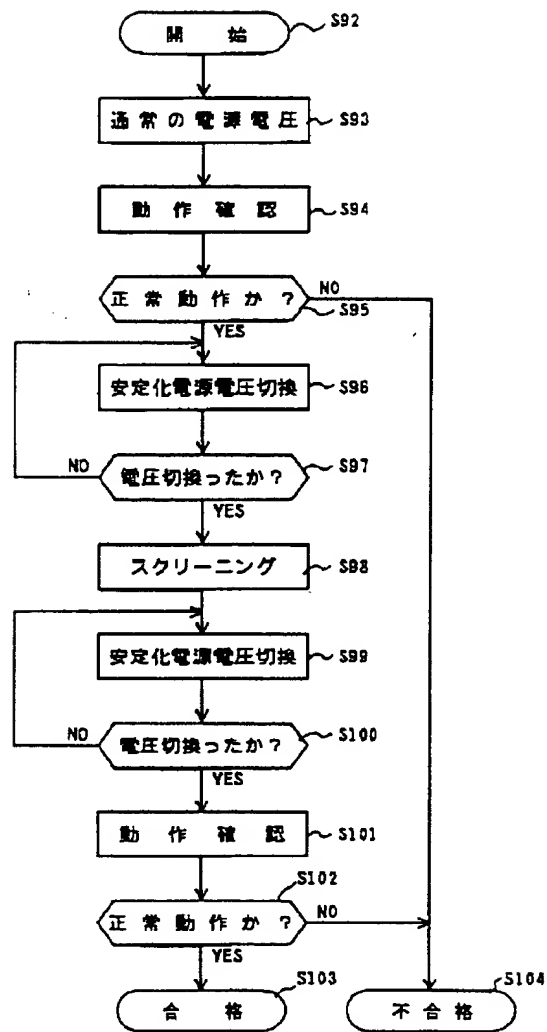


8



【図9】

図 9



フロントページの続き

(51)Int.Cl.<sup>5</sup> -  
 G 0 6 F 15/78  
 G 1 1 C 7/00

識別記号 庁内整理番号  
 5 1 0 G

F I

技術表示箇所

(12)

特開平7-287603

(72)発明者 石川 秀明  
茨城県勝田市大字高場2520番地 株式会社  
日立製作所自動車機器事業部内